

# Troubleshooting Plugins

## File Dependency - Bad External File References for Plugins

**“Plugins must maintain their own local copies of all the files they need to build, independent of the main OCPN source tree”**

Problem: In the CmakeLists.txt, uses direct references to the OpenCPN source tree, e.g.  
Code:

```
#include {OCPN Source Tree}/include/cutil.h
```

. .or.

```
../..src/wxJSON/jsonval.cpp
```

This is not allowed, and is sure to break sooner or later. Plugin source trees may take local copies of anything that they want from the OCPN tree. But the only thing that may be used dynamically is OCPNPlugin.h

What is happening here is that the master branch is evolving, as it should. And some change in the base code broke the Plugin source. So sorry, but that will happen often if Plugin devs don't follow the rules.

Said another way: Plugins must maintain their own local copies of all the files they need to build, independent of the main OCPN source tree. All you really need from the main tree is OCPNPlugin.h, and it need not be current. It only needs to be current up to the Plugin API revision supported by the Plugin in question.

Take a look at Dashboard. It contains local copies of all the NMEA decoding classes, and they are simply clones of the same files in the base code. Not links. That's the way our Plugin API works.

And another view: Unless a Plugin changes its API revision requirement, there *must* be no need to re-build a Plugin for a new base version. We promise backward compatibility this way.

## Tracing Bad (Illegal) External File References.

Go back to the version where it compiles properly, and trace the external file refs to opencpn base, then copy and change the path in the code

The easiest way to find the bad references would be to move the Plugin build tree completely out of the OCPN build tree. Then, the affected files won't be found by relative reference, and you will get a lot of build errors. Fix them one at a time by copying the files you need from an earlier version of OCPN, say 3.2.2. Put the copied files into the new Plugin build tree. Fix CMakeLists.txt. Straightforward, if tedious...

## Blacklist

Plugins, or versions of plugins, known to create problems for users, can be blacklisted.

This means they will not load, and a warning will be issued. For current list check “pluginmanager.h” line ~70.

From:  
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:  
[https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer\\_manual:plugins:plugin\\_api:troubleshooting\\_plugins](https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:plugins:plugin_api:troubleshooting_plugins)

Last update: **2017/01/13 22:37**

